



Práctica Evaluativa - Parcial I

Asignatura: Programación I

Tema: Estructuras Repetitivas con Listas

Caso 1 — Biblioteca escolar — Préstamos de libros

Caso 1 – Biblioteca escolar – Préstamos de libros

Enunciado / Descripción

La biblioteca escolar necesita un sistema de gestión sencillo para su catálogo de libros y las copias disponibles. Se pide desarrollar un programa con una interfaz basada en menú que utilice listas paralelas (una para títulos[] y otra para ejemplares[]). Cada título debe estar vinculado a su número correspondiente de copias utilizando el mismo índice en ambas listas. Se debe utilizar un bucle while para navegar por las opciones del menú hasta que el usuario elija salir.

Ejemplo:

- títulos[] = ["El Señor de los Anillos", "Orgullo y Prejuicio", "Matar un Ruiseñor"]
- ejemplares[] = [5, 3, 7]

En este ejemplo, "El Señor de los Anillos" tiene 5 copias, "Orgullo y Prejuicio" tiene 3 copias, y "Matar un Ruiseñor" tiene 7 copias.

Opciones del Menú:

- 1. Ingresar lista de títulos:
 - o Permite al usuario introducir los títulos de los libros en la biblioteca.
 - o Ejemplo: El usuario introduce "1984", "Rebelión en la Granja".
- 2. Ingresar lista de ejemplares disponibles (por título):
 - Permite al usuario introducir el número de copias disponibles para cada título de libro.
 - Ejemplo: Si el título es "1984", el usuario podría introducir "4" (lo que significa que hay 4 copias).
- 3. Mostrar catálogo con stock:
 - Muestra una lista de todos los títulos y el número de copias disponibles para cada uno.
 - Ejemplo de salida:
 - "El Señor de los Anillos: 5 copias"



- "Orgullo y Prejuicio: 3 copias"
- "Matar un Ruiseñor: 7 copias"
- 4. Consultar disponibilidad de un título específico:
 - o Permite al usuario introducir un título y ver cuántas copias están disponibles.
 - Ejemplo: El usuario introduce "Orgullo y Prejuicio", y el programa muestra "3 copias disponibles".
- 5. Listar agotados (0 ejemplares):
 - Muestra una lista de todos los títulos que tienen 0 copias disponibles.
- 6. Agregar título:
 - Permite al usuario añadir un nuevo título al catálogo y especificar el número inicial de copias.
- 7. Ver títulos agotados:
 - o Muestra una lista de los títulos con cero copias disponibles.
- 8. Actualizar ejemplares (préstamo/devolución):
 - Permite al usuario actualizar el número de copias cuando un libro es prestado (préstamo) o devuelto (devolución).
 - Ejemplo: Si alguien toma prestada una copia de "El Señor de los Anillos", el usuario puede actualizar el conteo de 5 a 4.
- 9. Ver catálogo:
 - o Muestra el catálogo entero de los títulos de libros.
- 10. Salir:
 - Sale del programa.



Entregables

El estudiante deberá subir el archivo del programa en lenguaje Python a la plataforma institucional. NO SUBIR UN REPOSITORIO DE GITHUB. SOLO SUBIR EL ARCHIVO.PY El código debe cumplir con:

- Todas las funcionalidades solicitadas reflejadas en el menú.
- Buena ejecución sin errores.
- Nomenclatura clara en el nombre de las variables.
- Legibilidad general y buenas prácticas de codificación.

Rúbrica de Evaluación

| Rúbrica de Evaluación Detallada para Gestión de Biblioteca (Préstamo de Libros) | | | | | |
|---|----------------------------------|------|---|--|--|
| Código | Criterio | Peso | Descripción Detallada | | |
| C1 | Correctitud Funcional | 50% | Qué evaluar específicamente en este caso: El correcto funcionamiento de las funcionalidades centrales del sistema. Agregar título: La capacidad de añadir nuevos libros al catálogo, asegurando que no se repitan los títulos. Consultar disponibilidad: Permite verificar cuántos ejemplares están disponibles para un título específico. Ver agotados: Permite listar los títulos que no tienen ejemplares disponibles (ejemplares = 0). Esto indica que todos los ejemplares están prestados o no hay ejemplares en el catálogo. Préstamo: Permite registrar el préstamo de un libro, disminuyendo la cantidad de ejemplares disponibles. Devolución: Permite registrar la devolución de un libro, aumentando la cantidad de ejemplares disponibles. Catálogo completo: Permite ver una lista de todos los títulos en el catálogo, con su respectiva cantidad de ejemplares disponibles. Manejar ejemplares = 0 y título inexistente sin errores: El programa debe manejar correctamente las situaciones en las que se intenta prestar un libro sin ejemplares disponibles, o cuando se intenta realizar una operación sobre un título que no existe en el catálogo, mostrando mensajes de error informativos y evitando que el programa falle. | | |
| C2 | Cumplimiento de Restricciones | 20% | Qué evaluar específicamente en este caso: El apego a las limitaciones de diseño impuestas. Sólo listas paralelas: El sistema debe implementarse utilizando únicamente listas paralelas para almacenar | | |



| | | | A DISTANCIA |
|--------|-----------------------------|------|---|
| Código | Criterio | Peso | Descripción Detallada |
| | | | los títulos y la cantidad de ejemplares. No se permite el uso de diccionarios, clases, u otras estructuras de datos. Conservar títulos con ejemplares = 0: El sistema debe mantener el registro de un libro aunque no haya ejemplares disponibles. El título no debe ser eliminado del catálogo. Sin diccionarios ni clases: Se evalúa la capacidad de resolver el problema utilizando las herramientas proporcionadas (listas paralelas), penalizando el uso de estructuras más complejas. Sincronía entre titulos[] y ejemplares[]: La cantidad de ejemplares en la posición i de la lista ejemplares[] debe corresponder al título en la posición i de la lista titulos[]. Se debe verificar que esta correspondencia se mantenga al añadir, prestar o devolver libros. |
| C3 | Interacción y Validación | 10% | Qué evaluar específicamente en este caso: La calidad de la interacción con el usuario y la integridad de los datos ingresados. Nombre no vacío: El programa debe validar que el nombre del título ingresado por el usuario no esté vacío. Cantidades enteras válidas: El programa debe validar que las cantidades ingresadas (ej., cantidad a prestar, cantidad a devolver) sean números enteros. Préstamo ≤ ejemplares: Al prestar un libro, el programa debe verificar que la cantidad a prestar no sea mayor que la cantidad de ejemplares disponibles. Devolución ≥1: Al registrar una devolución, el programa debe verificar que la cantidad a devolver sea mayor o igual a 1. Título inexistente: Al intentar prestar o devolver un libro, o consultar su disponibilidad, el programa debe verificar que el título exista en el catálogo. Menú persistente: El menú principal debe permanecer visible y funcional hasta que el usuario seleccione la opción de salir. Opciones inválidas: El programa debe manejar las opciones inválidas del menú (ej., ingresar letras en lugar de números) mostrando mensajes de error claros y apropiados. |
| C4 | Estructura y Legibilidad | 10% | Qué evaluar específicamente en este caso: La organización, claridad y mantenibilidad del código. • Variables descriptivas: Los nombres de las variables |



| Código | Criterio | Peso | Descripción Detallada |
|--------|--|------|--|
| | | | deben ser significativos y reflejar el propósito de la variable. Flujo claro: El código debe seguir un flujo lógico y fácil de entender. Sangría: El código debe tener una indentación consistente para mejorar la legibilidad. Evitar duplicación: Se debe evitar la duplicación innecesaria de código mediante el uso de funciones. Mensajes consistentes: Los mensajes mostrados al usuario deben ser coherentes y fáciles de entender. |
| C5 | Casos de Prueba / Datos de Ejemplo | 5% | Qué evaluar específicamente en este caso: La cobertura de los casos de prueba, incluyendo casos normales y situaciones límite. Pruebas normales y borde: Incluir pruebas para las funciones principales (agregar, consultar, prestar, devolver) con valores típicos, así como con valores límite (ej., intentar prestar 0 libros). Ejemplares = 0: Probar el sistema con un título que no tiene ejemplares disponibles. Préstamo mayor que disponible: Intentar prestar una cantidad de libros mayor a la disponible. Título inexistente: Intentar prestar o devolver un libro que no existe en el catálogo. Etc.: Se deben incluir otros casos de prueba para cubrir diferentes escenarios y situaciones límite. |
| C6 | Gestión de Casos Borde | 5% | Qué evaluar específicamente en este caso: El manejo de situaciones excepcionales que pueden ocurrir. Préstamo con 0 ejemplares: Intentar prestar un libro cuando no hay ejemplares disponibles. Préstamo mayor que stock: Intentar prestar un número negativo de libros, o un número excesivamente grande. Devolución inválida: Intentar devolver un número negativo de libros. Título vacío o inexistente: Intentar realizar una operación (préstamo, devolución, consulta) sobre un título que tiene un nombre vacío o que no existe en el catálogo. |

Descriptores globales por nivel

Excelente (90–100): Cumple todos los criterios sin fallas; maneja casos borde exhaustivamente; interacción robusta; código claro y consistente.

Muy Bueno (80–89): Cumple la mayoría; pequeños desajustes no críticos; validación adecuada; estilo mayormente claro.



Aprobado (60–79): Cumple los básicos; algunos fallos funcionales menores o validación incompleta; legibilidad aceptable.

Insuficiente (<60): Incumple criterios clave; errores que impiden el uso; omite restricciones solicitadas.

Penalizaciones y observaciones

- -30% si utiliza estructuras prohibidas (diccionarios, clases, etc.).
- -10% si no conserva ítems con cantidad 0 (elimina de una lista y no de la otra).
- -10% por ausencia total de validación de entradas.
- -5% por mensajes ininteligibles o menú que no persiste.

Notas

- Asignar puntajes parciales por criterio y calcular la nota final ponderada.
- Compartir la rúbrica con estudiantes antes de la evaluación.
- Mantener consistencia entre enunciados, resultados de aprendizaje y criterios de evaluación.

Reglas Lógicas Adaptadas (Sin Funciones, Excepciones, Diccionarios ni Estructuras Avanzadas):

- C1 Correctitud Funcional:
 - Agregar título:
 - Comprobación de la lista titulos[] y ejemplares[]:
 - Después de la sección de código que se espera que agregue un título:
 - Verificar que la longitud de la lista titulos[] haya aumentado en 1.
 - Verificar que la longitud de la lista ejemplares[] haya aumentado en 1.
 - Verificar que el nuevo título (simulado) esté presente en la lista titulos[].
 - Verificar que la cantidad inicial de ejemplares (simulada) esté presente en la lista ejemplares[].
 - Evitar títulos duplicados:
 - Ejecutar el código que agrega un título duplicado.
 - Verificar que la longitud de la lista titulos[] no haya cambiado.
 - Verificar que no se agregue el título duplicado a la lista titulos[].
 - Verificar (analizando la salida estándar) que se imprima un mensaje (si se espera) indicando el error.



- Consultar disponibilidad:
 - Búsqueda del título:
 - Ejecutar la sección de código que busca un título específico.
 - Capturar la salida estándar del programa.
 - Verificar que la salida estándar contenga la cantidad correcta de ejemplares disponibles para ese título.
 - Título inexistente:
 - Ejecutar la sección de código que busca un título que no existe.
 - Capturar la salida estándar del programa.
 - Verificar que la salida estándar contenga un mensaje indicando que el título no se encuentra.
- Ver agotados:
 - Lista de agotados:
 - Ejecutar la sección de código que muestra la lista de títulos agotados.
 - Capturar la salida estándar del programa.
 - Verificar que la salida estándar contenga la lista correcta de títulos con ejemplares[i] == 0, en el formato esperado (si hay un formato específico requerido).
- C2 Cumplimiento de Restricciones:
 - Sólo listas paralelas: (Misma lógica que antes)
 - Analizar el código fuente (manualmente o con expresiones regulares simples) para verificar que no se utilizan diccionarios, clases, etc.
 - Sincronía entre listas: (Misma lógica que antes)
 - Después de cada operación, verificar que las listas titulos[] y ejemplares[] tienen la misma longitud.
 - Comprobar visualmente (si no hay otra opción) que los índices de las listas se corresponden. Si el código imprime las listas en un formato predecible, se puede analizar la salida para verificar la correspondencia.
- C3 Interacción y Validación:
 - Nombre no vacío:
 - Simular la entrada de un nombre vacío.
 - Capturar la salida estándar del programa.



- Verificar que la salida estándar contenga un mensaje de error.
- Verificar que no se haya agregado un nuevo título.

Cantidades enteras válidas:

- Simular la entrada de un valor no entero.
- Capturar la salida estándar del programa.
- Verificar que la salida estándar contenga un mensaje de error.
- Verificar que no se haya agregado un nuevo título.

o Préstamo ≤ ejemplares:

- Ejecutar la sección de código que permite prestar un libro.
- Simular la entrada de una cantidad a prestar mayor que la disponible.
- Capturar la salida estándar del programa.
- Verificar que la salida estándar contenga un mensaje de error.
- Verificar que la cantidad de ejemplares disponibles no haya cambiado.



